



IBM Software Group

# Using MQTT for connection of mobile applications to IBM MessageSight and enterprise messaging systems

Valerie Lampkin (vlampkin@us.ibm.com)  
Middleware Technical Resolution Specialist  
April 10, 2014



WebSphere® Support Technical Exchange



# Agenda

- **Proliferation of mobile devices creating the “Internet of Things”**
- **MQ Telemetry Transport (MQTT): low bandwidth protocol utilizing Pub/Sub architecture with Quality of Service (QoS) for assured delivery**
- **WebSphere MQ (WMQ) Telemetry “Extended Reach”**
- ***MessageSight handles “big data” coming from millions of remote devices.***
- ***Connecting Mobile Applications to IBM® MessageSight and WebSphere MQ Using MQTT***
- ***Examples of mobile user experience with MQTT and MessageSight***
- ***How To connect MessageSight to a WebSphere MQ Queue Manager (QMGR).***
- ***How to connect mobile app to MessageSight which then can access the backend enterprise***

# Internet of Things



- The proliferation of intelligent devices is changing the way people interact.
- A technology-enabled world is more interconnected than ever before.
- Large amounts of data from mobile devices and remote sensors are constantly being collected and transmitted.
- Creates a need for a system that can process Big Data and provide real-time analysis and response.

# MQTT Protocol

- MQ Telemetry Transport (MQTT) is a simple publish/subscribe lightweight messaging protocol.
- It is open source and royalty-free, allowing easy adaptation for a wide variety of devices.
- Ideal for constrained environments where network bandwidth is low and when remote devices may have limited processing capability.
- Eclipse Paho Project: a collaboration of IBM, Eurotech and other entities to provide open source tools and protocols as part of project to simplify development of Machine to Machine (M2M) solutions.
- OASIS® MQTT Technical Committee: In 2013, an OASIS technical committee was formed to produce a standard for MQTT with requirements for enhancements, documented usage examples and best practices.

More information available at these websites:

[www.eclipse.org/paho/](http://www.eclipse.org/paho/)

[www.mqtt.org](http://www.mqtt.org)

[www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=mqtt](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt)



# MQTT compared to HTTP

	MQTT	HTTP
<b>Design orientation</b>	Data centric	Document centric
<b>Pattern</b>	Publish/subscribe	Request/response
<b>Complexity</b>	Simple	More complex
<b>Message size</b>	Small, with a compact binary header just two bytes in size	Larger, partly because status detail is text-based
<b>Service levels</b>	Three quality of service settings	All messages get the same level of service
<b>Extra libraries</b>	Libraries for C (30 KB) and Java (100 KB)	Depends on the application (JSON, XML), but typically not small
<b>Data distribution</b>	Supports 1 to zero, 1 to 1, and 1 to <i>n</i>	1 to 1 only

# Features of MQTT Protocol

- A Publish/Subscribe messaging model that facilitates one-to-many distribution. Sending applications or devices don't need to know anything about the receiver, not even its address.
- Ideal for constrained networks (low bandwidth, high latency, data limits, fragile connections). MQTT message headers are kept as small as possible; the fixed header is just 2 bytes. Its on demand, push-style message distribution keeps network utilization low.
- Multiple service levels allow flexibility in handling different types of messages. Developers can designate that messages will be delivered “at most once”, “at least once”, or “exactly once”.

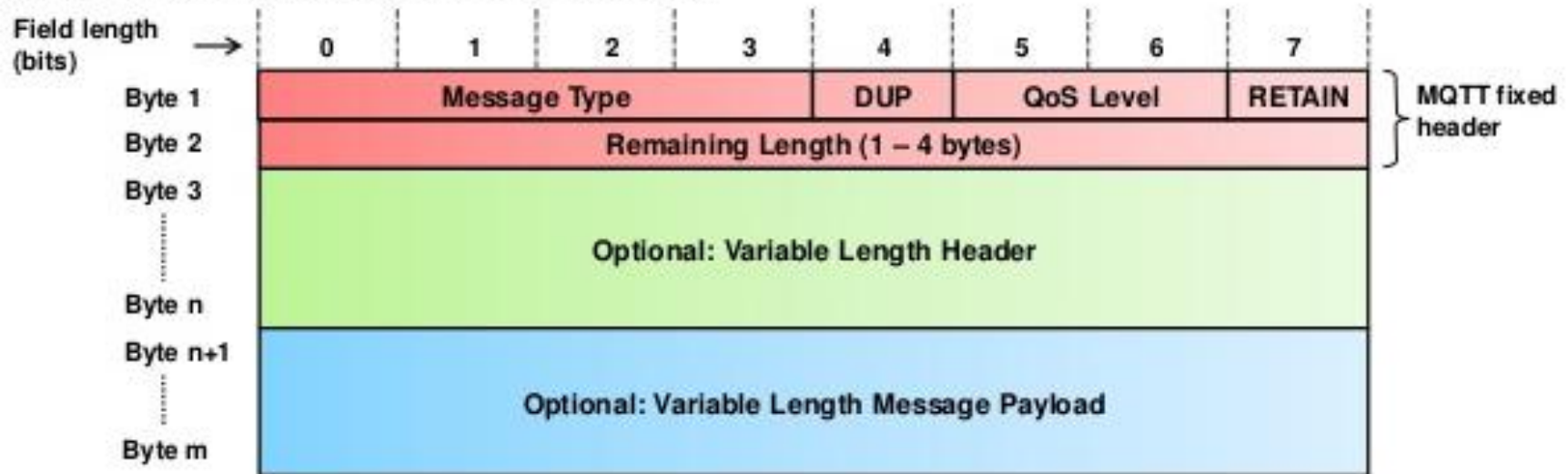
## Features of MQTT Protocol (Continued)

- Designed specifically for remote devices with little memory or processing power. Minimal headers, a small client footprint and limited reliance on libraries make MQTT ideal for constrained devices.
- Easy to use and implement with a simple set of command messages. Many applications of MQTT can be accomplished using just CONNECT, PUBLISH, SUBSCRIBE and DISCONNECT.
- Built-in support for loss of contact between client and server. The server is informed when a client connection breaks abnormally, allowing the message to be re-sent or preserved for later delivery.

# MQTT Message Format

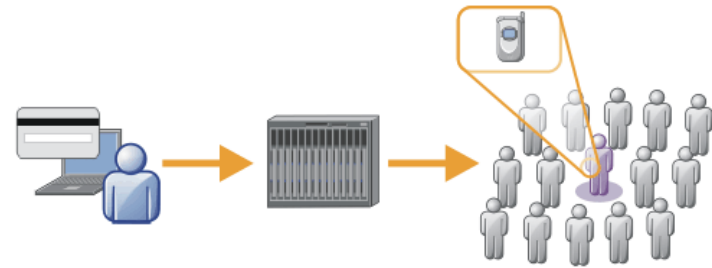
- A Mandatory fixed-length header (2 bytes)
- The other fields are optional allowing MQTT to be optimized for constrained unreliable networks.
- Optional message-specific variable length header and message payload.

MQTT uses network byte and bit ordering.

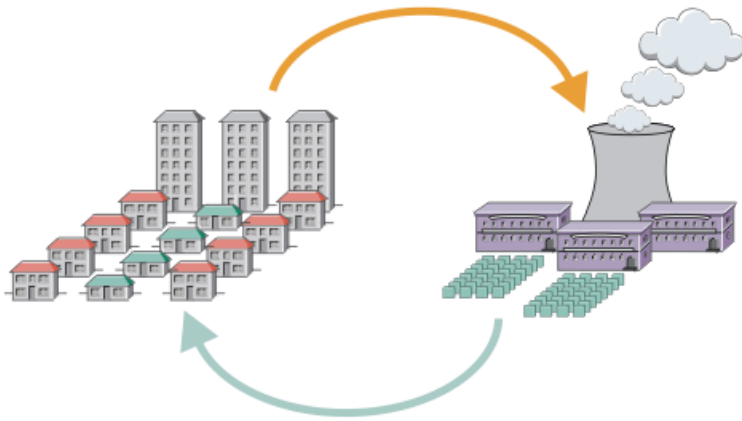




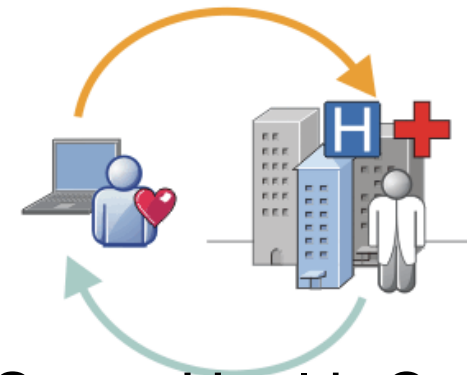
# Examples of MQTT implementation



Smart Banking



Smart Energy Monitoring



Smart Health Care

# Concepts of MQTT

- **Publish/Subscribe**

The MQTT protocol is based on the principle of publishing messages and subscribing to topics, which is typically referred to as a PUBLISH/SUBSCRIBE model. Clients can subscribe to topics that pertain to them and thereby receive whatever messages are published to those topics. Or clients can publish messages to topics, thus making them available to all subscribers to those topics.



# Concepts of MQTT (continued)

## Topics & subscriptions

Messages in MQTT are published to topics, which can be thought of as subject areas. Clients, in turn, sign up to receive particular messages by subscribing to a topic. Subscriptions can be explicit, which limits the messages received to the specific topic at hand, or use wildcard designators (+ and #) to receive messages across a variety of related topics.

- **Retained messages**

With MQTT, the server keeps the message even after sending it to all current subscribers. If a new subscription is submitted for the same topic, any retained messages are then sent to the new subscribing client.



# Concepts of MQTT (continued)

## ■ Clean sessions & durable connections

When an MQTT client connects to the server, it sets the clean session flag. If the flag is set to true, then all of the client's subscriptions are removed when it disconnects from the server. If the flag is set to false, then the connection is treated as durable, and the client's subscriptions remain in effect after any disconnection. In this event, subsequent messages that arrive carrying a high QoS designation are stored for delivery once the connection is re-established.



# Concepts of MQTT (continued)

## ■ Last Will and Testament

When a client connects to a server, it can inform the server that it has a *will*, or a message that should be published to a specific topic or topics in the event of an unexpected disconnect. This is particularly useful in alarm or security settings where system managers must know immediately when a remote sensor has lost contact with the network.

## ■ Qualities of Service

MQTT defines three Quality of Service (QoS) levels for message delivery (more details on next slide)



# QoS – Quality of Service

## Three qualities of service for message delivery:

- QoS = 0 "At most once", messages are delivered according to the best efforts of TCP/IP network. Message loss or duplication can occur. A response is not expected and no retry defined in the protocol
- QoS=1 "At least once", where messages are assured to arrive but duplicates may occur.
- QoS =2 "Exactly once", where message are assured to arrive exactly once.

Note: Performance impact in terms of network traffic and processing power for greater QoS



# IBM WebSphere MQ Telemetry / Extended Reach

- MQ Telemetry was added as an installable feature of IBM WebSphere MQ 7.0.1 but is now fully integrated as part of WebSphere MQ version 7.1 and 7.5
- IBM's WebSphere MQ Telemetry is a feature of WebSphere MQ that extends the universal messaging backbone with the MQTT protocol to a wide range of remote sensors, actuators and telemetry devices. It enables instrumented devices -- virtually anywhere in the world -- to connect to each other, and to enterprise applications, web services and decision makers with WebSphere MQ.

# IBM WebSphere MQ Telemetry / Extended Reach (continued)

The WebSphere MQ Telemetry service (MQXR) is a TCP/IP listener that is part of a WebSphere MQ queue manager. MQ Telemetry clients are remote devices and applications that use the MQTT protocol to send and receive messages with the WebSphere MQ Queue Manager. From the WebSphere MQ Queue Manager, the messages can be exchanged with other messaging applications such as similar telemetry applications, MQ API, JMS™, or enterprise messaging applications such as WebSphere Application Server and Message Broker (now called IIB – IBM Integration Bus v9)



# IBM MessageSight

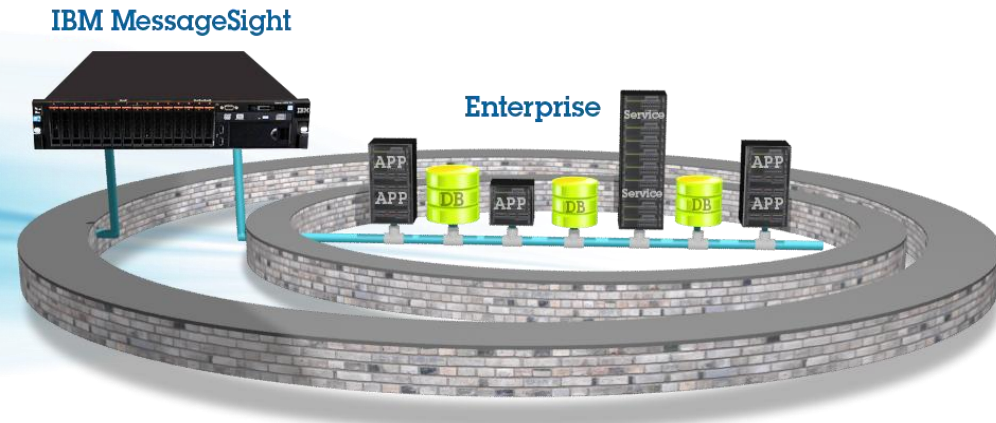


IBM MessageSight is an appliance-based MQTT server that can connect a massive number of MQTT clients at the same time, and deliver the performance and scalability needed to accommodate the ever growing multitude of mobile devices and sensors. It supports the MQTT version 3.1 protocol, and MQTT over the WebSocket protocol

“Easy-to-deploy” -- has been shown to go from unpacking the box to up and running in less than 30 minutes in this YouTube video:

<http://www.youtube.com/watch?v=RZJbqWLj1Zc>

# IBM MessageSight



- A low-latency, reliable and scalable messaging server – designed specifically for **M2M** and **Mobile** scenarios
- The DMZ-ready appliance form factor provides **strong security** and **easy deployment**
- Enables the next generation of applications with **event-driven, near-real-time communications**

# Features of IBM MessageSight

- **High Availability**

A pair of appliances may be connected for a Hot-Warm setup.

- **Developer Friendly**

Simple yet powerful APIs: connect, subscribe, publish, disconnect

- **Optimized for Wireless**

- ▶ Open Royalty Free Specifications being standardized by OASIS

- ▶ Efficient two-way messaging **protocol** designed for constrained devices, low-bandwidth, unreliable networks



# Features of IBM MessageSight (continued)

## ■ **Compatibility/Easy to Integrate**

- ▶ Integrate with Application Servers such as WAS using Java™ Connector Architecture (JCA)
- ▶ Built-in Connectivity with WebSphere MQ
- ▶ IBM Integration Bus (broker) support through JMS nodes
- ▶ InfoSphere Streams can consume data from MessageSight

## ■ **Security**

- ▶ Support for TLS/SSL security
- ▶ Authentication by userid/password or by certificate
- ▶ Easy to configure security profiles, each Endpoint (port) can be configured

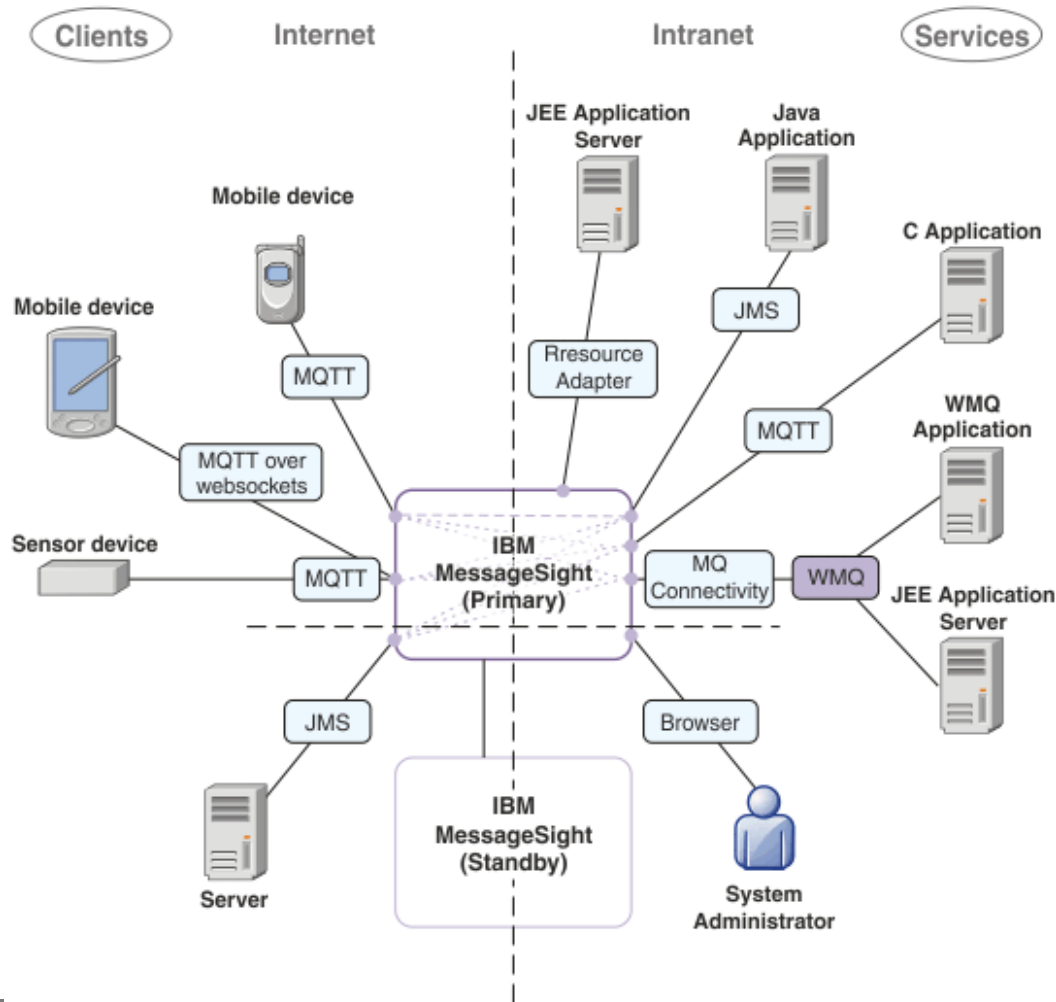


# Features of IBM MessageSight

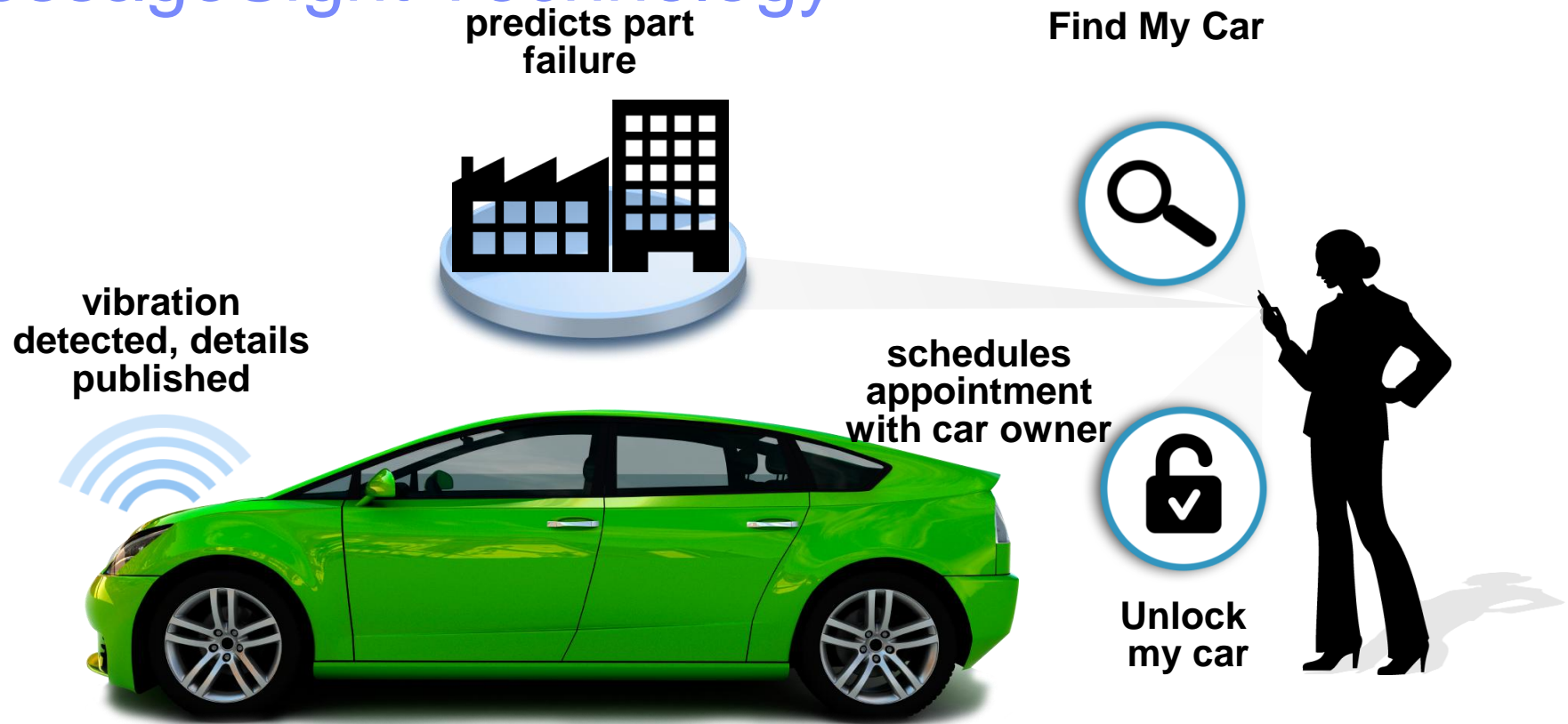
- **Easy Administration**
  - ▶ Up and running in less than 30 minutes
  - ▶ Web User Interface allows monitoring
- **Massive Scalability**
- One appliance can handle more than 1 million concurrent connections.
  - ▶ V1.1 offers shared subscriptions which allows applications to share and balance workload across multiple processes (e.g. workload can be split across multiple instance of IBM Integration Bus)



# IBM MessageSight Appliance Connectivity



# “Connected Car” example of MQTT and MessageSight Technology

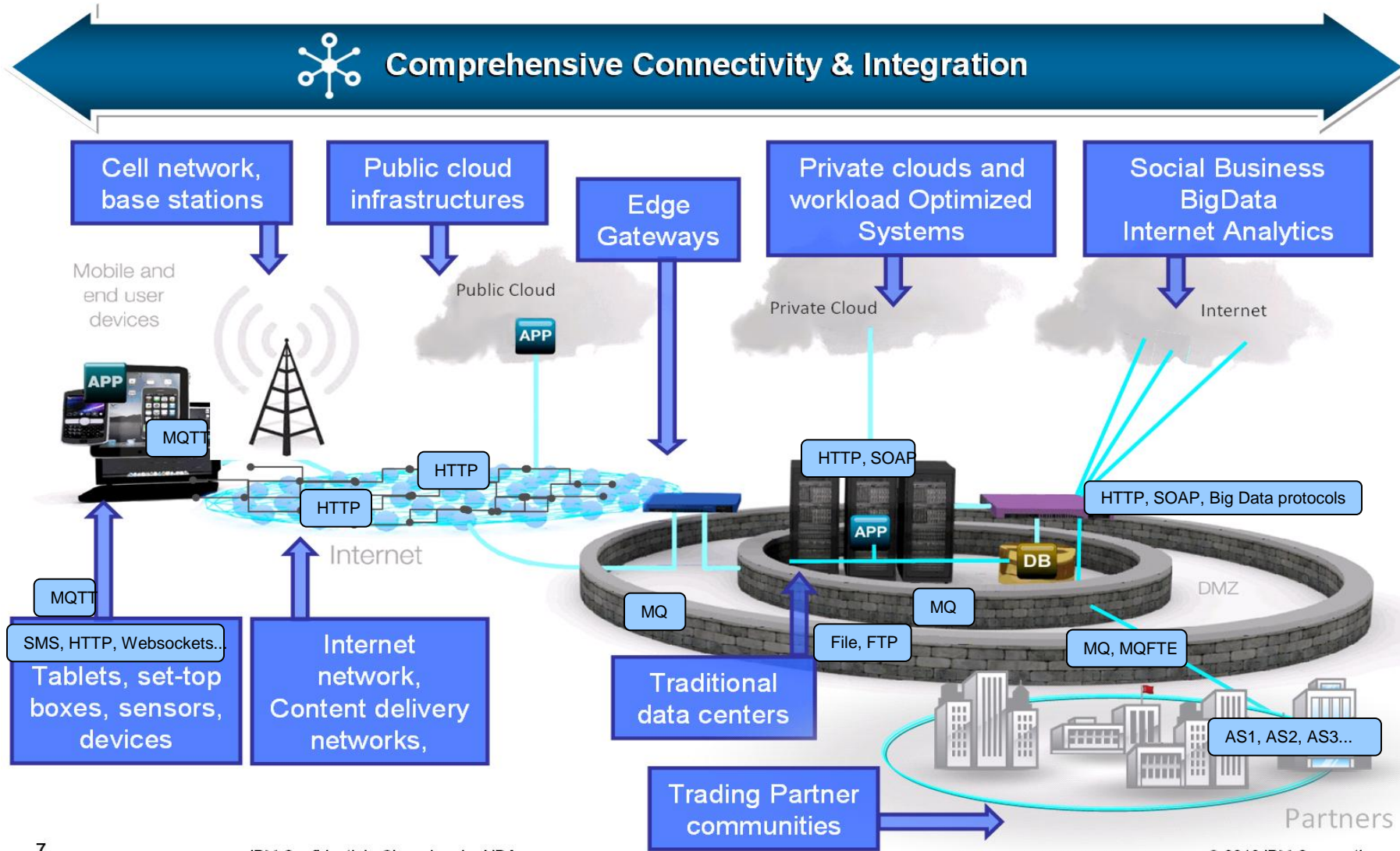


SmartCar

<http://m2m.demos.ibm.com/autoremote/>



# Connectivity Extending from Enterprise to "the world"





# Connecting MessageSight to Enterprise Systems



**You can configure the IBM MessageSight appliance to link with new or existing WebSphere Queue Managers using the MQ Connectivity feature. The MQ Connectivity feature can be used from MQ v7.1 onwards.**

# Connecting MessageSight to Enterprise Systems

- To connect the MessageSight appliance to MQ, you create a svrconn channel on the queue manager then you configure destination mapping rules. This can be accomplished with the **runmqsc** command on the server where the queue manager is running. Example:

```
$ runmqsc MQTT_QMGR
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2011. ALL RIGHTS  
RESERVED.
```

```
Starting MQSC for queue manager MQTT_QMGR.
```

```
define chl(IMA.SVRCONN) chltype(SVRCONN)
```

```
1 : define chl(IMA.SVRCONN) chltype(SVRCONN)
```

```
AMQ8014: WebSphere MQ channel created.
```

# MQ Connectivity

**After the SVRCONN Channel is created, the MQ Connectivity between MessageSight and the WebSphere MQ Queue Manager can be configured with a few steps:**

- Log into the MessageSight appliance with a browser using https IP address and port number. Sign in as administrator

IBM® MessageSight™

\* User ID:

\* Password:

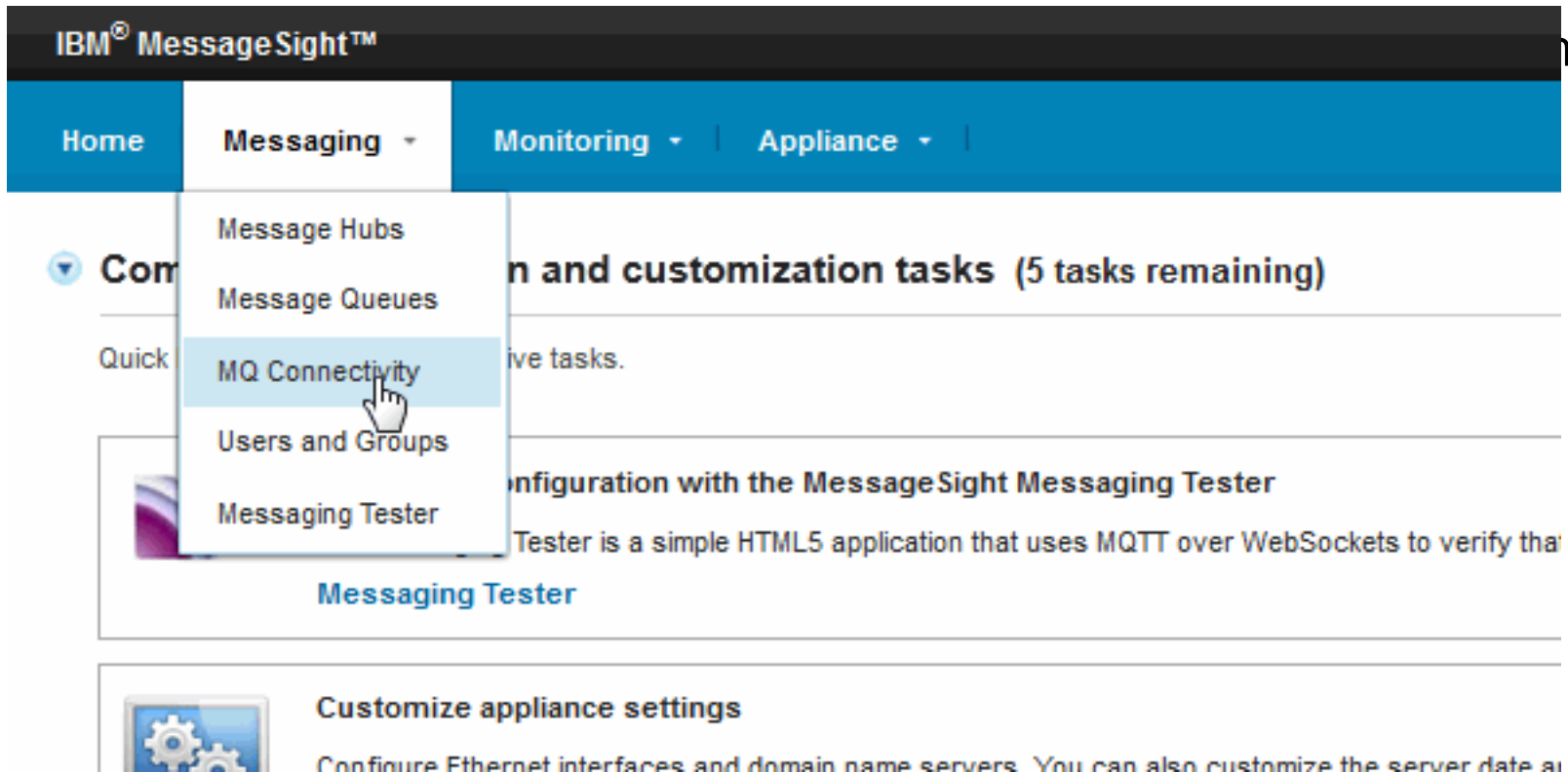
**Log in**

IBM, the IBM logo, and MessageSight are trademarks of IBM Corporation, registered in many jurisdictions worldwide. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Java  
COMPATIBLE

Licensed Materials - Property of IBM. 5725-F96 © Copyright IBM Corporation 2012, 2013.

# Connecting MessageSight to MQ



The screenshot displays the IBM MessageSight web application interface. At the top, the header reads "IBM® MessageSight™". Below the header is a navigation bar with tabs for "Home", "Messaging", "Monitoring", and "Appliance". The "Messaging" tab is selected, and its dropdown menu is open, showing options: "Message Hubs", "Message Queues", "MQ Connectivity" (highlighted with a mouse cursor), "Users and Groups", and "Messaging Tester".

Below the navigation bar, the main content area features a "Configuration and customization tasks (5 tasks remaining)" section. A "Quick" link is visible. A "Messaging Tester" link is also present, with a description: "Configuration with the MessageSight Messaging Tester" and "The Tester is a simple HTML5 application that uses MQTT over WebSockets to verify that".

At the bottom, there is a "Customize appliance settings" section with a gear icon and the text: "Configure Ethernet interfaces and domain name servers. You can also customize the server date and time".

# Connecting MessageSight to MQ

The Queue Manager Connection Properties are presented for you to specify the connection details for a WebSphere MQ queue manager. Click the green + sign to add queue manager connection.

IBM® MessageSight™ Status ▾ admin ▾

Home Messaging ▾ Monitoring ▾ Appliance ▾

## MQ Connectivity

Configure connections to one or more WebSphere MQ Queue Managers.

### Queue Manager Connection Properties

System Administrators and Messaging Administrators can define, edit, or delete information about how the appliance connects to the Queue Managers.

+ ✕ ✎ Filter  ↻

Name	Queue Manager	Connection Name	Channel Name	SSL CipherSpec
No items to display				

Total: 0 Selected: 0 ◀ 1 ▶ 10 25 50 100 ⚙

## Connecting MessageSight to MQ (continued)

Next step: in the “Add Queue Manager Connection” window, fill in the name of the connection, queue manager, connection info and the svrconn channel name (the name of the channel created earlier).

The screenshot shows the IBM MessageSight web interface. The top navigation bar includes 'Home', 'Messaging', 'Monitoring', and 'Appliance'. The main content area is titled 'MQ Connectivity' and contains a table for 'Queue Manager Connections'. A modal dialog box titled 'Add Queue Manager Connection' is open, displaying the following fields:

- Name: MQConnectivity
- Queue Manager: MQTT\_QMGR
- Connection Name: 9.123.45.67(1414)
- Channel Name: IMA.SVRCONN
- SSL CipherSpec: (empty)

At the bottom of the dialog are three buttons: 'Save', 'Test connection', and 'Cancel'.

## Security Considerations Connecting MessageSight to MQ

- The ID you use for MessageSight must have proper authority to access the MQ objects. Improper security may result in errors in MQ logs

*AMQ5653: The user 'admin' is not defined.*

*EXPLANATION: The system call getpwnam("admin") failed with errno -1.*

*ACTION: Create the user 'admin' and retry the operation.*

- The authorities may be granted with setmqaut commands

```
setmqaut -m MQTT_QMGR -t qmgr -g msadmin +connect +dsp +inq
```

```
setmqaut -m MQTT_QMGR -t queue -n SYSTEM.DEFAULT.MODEL.QUEUE -g  
msadmin +dsp +get
```

```
setmqaut -m MQTT_QMGR -t queue -n SYSTEM.ADMIN.COMMAND.QUEUE -g  
msadmin +dsp +put
```

```
setmqaut -m MQTT_QMGR -t queue -n SYSTEM.IMA.* -g msadmin +crt +put +get  
+browse
```

```
setmqaut -m MQTT_QMGR -t queue -n SYSTEM.DEFAULT.LOCAL.QUEUE -g  
msadmin +dsp
```

# Destination Mapping Rules

While on the MQ Connectivity tab of the Web UI, you can create the Destination Mapping Rules.

Click the green + under the “Destination Mapping Rules”. The “Add Destination Mapping Rule” shows you how to enter the rule information. The drop down box for Rule Type allows you to select how you want to map between topics and queues.

IBM® MessageSight™

## Add Destination Mapping Rule

Destination mapping rules define the direction in which messages are moved, and the nature of t

\* Name: [?](#)

Rule Type:  ▼

\* Source: [?](#)

\* Destination: [?](#)

\* Max Messages: [?](#)

Retained Messages: [?](#)

Enabled:

\* Associated queue management

Associated	Name
<input type="checkbox"/>	ITSO_IMA_C...

- MessageSight topic to MQ queue
- MessageSight topic to MQ topic
- MQ queue to MessageSight topic
- MQ topic to MessageSight topic
- MessageSight topic subtree to MQ queue
- MessageSight topic subtree to MQ topic
- MessageSight topic subtree to MQ topic subtree
- MQ topic subtree to MessageSight topic
- MQ topic subtree to MessageSight topic subtree
- MessageSight queue to MQ queue
- MessageSight queue to MQ topic
- MQ queue to MessageSight queue
- MQ topic to MessageSight queue
- MQ topic subtree to MessageSight queue



## Destination Mapping Rules

You also must check the box *Associated queue manager connections* to associate the Destination Mapping rule with the appropriate queue manager connection. After a rule is created and enabled, messages are routed as configured.

The screenshot shows the 'Add Destination Mapping Rule' configuration page in the IBM MessageSight interface. The page includes a sidebar with navigation options like 'Home', 'Destin', 'System', 'Rules m', and 'SSL Ke'. The main content area is titled 'Add Destination Mapping Rule' and contains the following configuration fields:

- Name:** TestMap
- Rule Type:** MessageSight topic to MQ queue
- Source:** (empty field)
- Destination:** (empty field)
- Max Messages:** 5000
- Retained Messages:** None
- Enabled:**
- Associated queue manager connections:** A table with one entry:

Associated	Name	Connection Name
<input checked="" type="checkbox"/>	TestConnection	9.12.5.184(1414)

# Showing destination mapping rules with command line

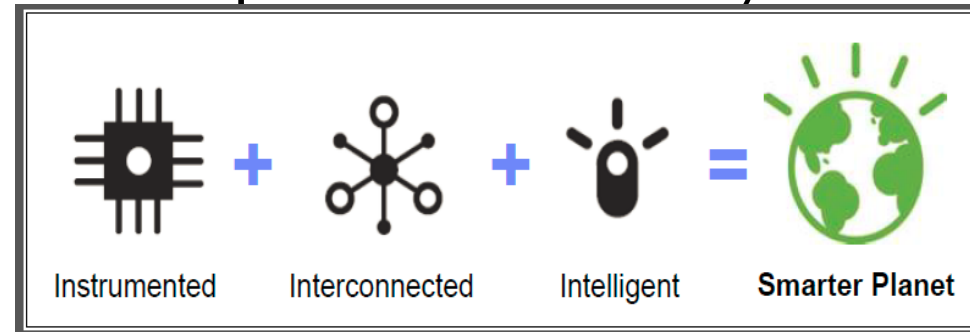
```
Console> imaserver list DestinationMappingRule
ITSO_IMA_QM_MAP_01
ITSO_IMA_QM_MAP_02
Console> imaserver show DestinationMappingRule "Name=ITSO_IMA_QM_MAP_01"
Name = ITSO_IMA_QM_MAP_01
QueueManagerConnection = ITSO_IMA_CONN
RuleType = 1
Source = /root/publication/temperature
Destination = QLOCAL.TEMPERATURE
MaxMessages = 5000
Enabled = True
RetainedMessages = None
Console> imaserver show DestinationMappingRule "Name=ITSO_IMA_QM_MAP_02"
Name = ITSO_IMA_QM_MAP_02
QueueManagerConnection = ITSO_IMA_CONN
RuleType = 3
Source = QLOCAL.TEMPERATURE.MONITOR
Destination = /root/publication/temperature/monitor
MaxMessages = 5000
Enabled = True
RetainedMessages = None
```

# Connecting MQTT Remote Devices

There are devices other than mobile phones which may connect via MQTT

Sensors, Actuators, RFID devices

Port 1883 is default for MQTT



Many different MQTT open source clients available

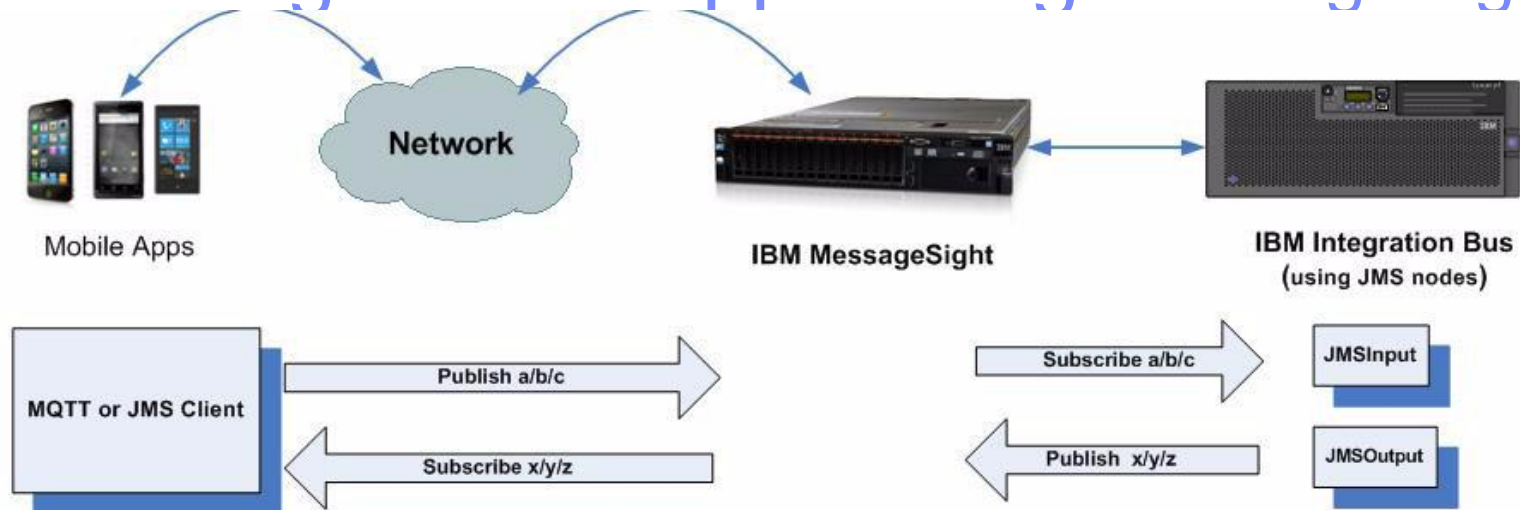
C, Java, JavaScript™, LotusScript®, .NET, Perl, Python, REXX, and more

YouTube video showing configuration of MessageSight to power a MQTT 'keyfob' application which uses pub/sub to control a simulated car

**IBM MessageSight: Powering an MQTT application**

<https://www.youtube.com/watch?v=-2kk6db7E7w#t=30>

# Connecting Mobile Apps using MessageSight



Mobile applications can utilize MessageSight to communicate with existing Enterprise Messaging systems such as the IBM Integration Bus. In the diagram above, the mobile applications publish on the request topics and subscribe to the related response topics. The JMS nodes in the IBM Integration Bus message flows subscribe to the request topics and publish on the response topics.

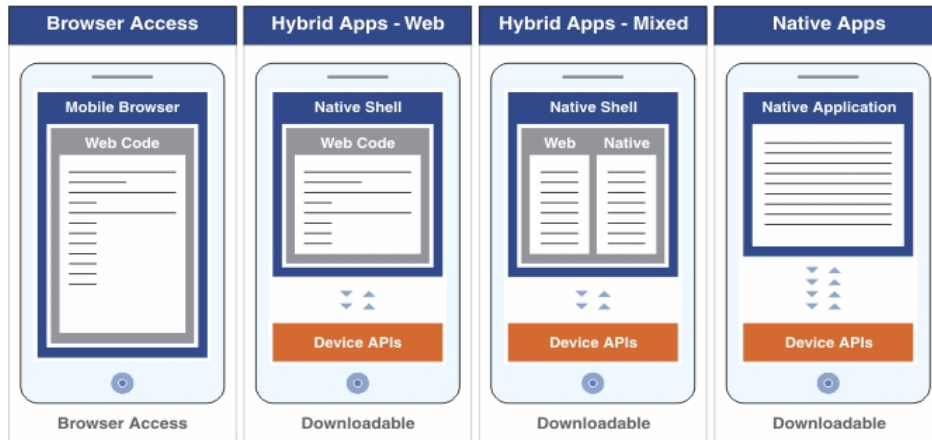
The mobile applications use MQTT to publish requests and subscribe to responses. The IBM Integration Bus message flows are using JMS to subscribe to requests and to publish responses. IBM MessageSight does the required protocol bridging between MQTT and JMS.

# Mobile Applications

## Factors to Consider when Developing Mobile Applications

- **Battery Life**
  - ▶ Applications should be designed to consume the least resources possible so that battery life is preserved
- **Processing Capacity**
  - ▶ Applications should be simple in terms of processing to speed response time
- **Network Optimization**
  - ▶ Mobile devices should have network connectivity options (choices that vary in speed, range, battery consumption and cost to the user). Application should be able to switch between available communication options seamlessly allowing for continued functionality in a transparent manner when the network connection changes (i.e. switch to 3G/4G when WiFi is out of range) .
- **Platform Capabilities**
  - ▶ Requirements to build application can vary from one platform to another. Usually the platform owner provides a set of APIS. With MQTT, freely available implementations of the protocol allow developers to concentrate on application functionality and use the MQTT protocol as a service.

# Mobile Applications



## Different Approaches for Developing Mobile Applications

- **Mobile Browser:** Web applications are written entirely in HTML5, CSS, and JavaScript code. These apps are run by the mobile browser and are cross-platform by default.
- **Hybrid Apps (Web/Mixed):**
  - ▶ **Web:** source code of the application consists of web code run within a native container and consists of native libraries.
  - ▶ **Mixed:** developer augments the web code with native language to create unique features and access native APIs that are not available in JavaScript
- **Native Apps:** platform-specific and requires expertise and knowledge of the platform

# IBM Worklight

IBM Worklight consists of five core components. Each component provides a specific set of functions and supports a different stage in the lifecycle of a mobile application:

- **Worklight Studio:** An Eclipse-based development environment with a rich set of tools for developing cross-platform applications
- **Worklight Device Runtime:** A runtime environment with consistent APIs that can work across different mobile platforms to access native device functionality and integrate with existing services and security
- **Worklight Server:** A runtime server that enables secure access to enterprise services, application version management, unified push notifications, and direct application updates
- **Worklight Console:** A web-based user interface for real-time analytic, push notification authoring, and mobile application version management
- **Worklight Application Center:** A private, cross-platform mobile application store that is tailored for the specific needs of an application development team

*IBM Worklight made updates to support Apple® iOS7 on the day it was announced!*

# IBM Worklight

- IBM Worklight leverages open standards, such as HTML, CSS, and JavaScript, and solutions such as Apache Cordova, Eclipse Foundation and the Android and Apple SDKs providing flexibility for implementing mobile applications
- IBM Worklight API uses Apache open source Cordova project (also known as PhoneGap) to bridge JavaScript to native Java code required to interface with Android platform and device hardware.

The Worklight Client API lets the mobile application JavaScript code use Android platform specific user interface controls and access device specific hardware.
- The application itself can be developed in HTML and JavaScript, and the Apache Cordova (PhoneGap) libraries provided with Worklight are used to interface with native code that implements the MQTT protocol.

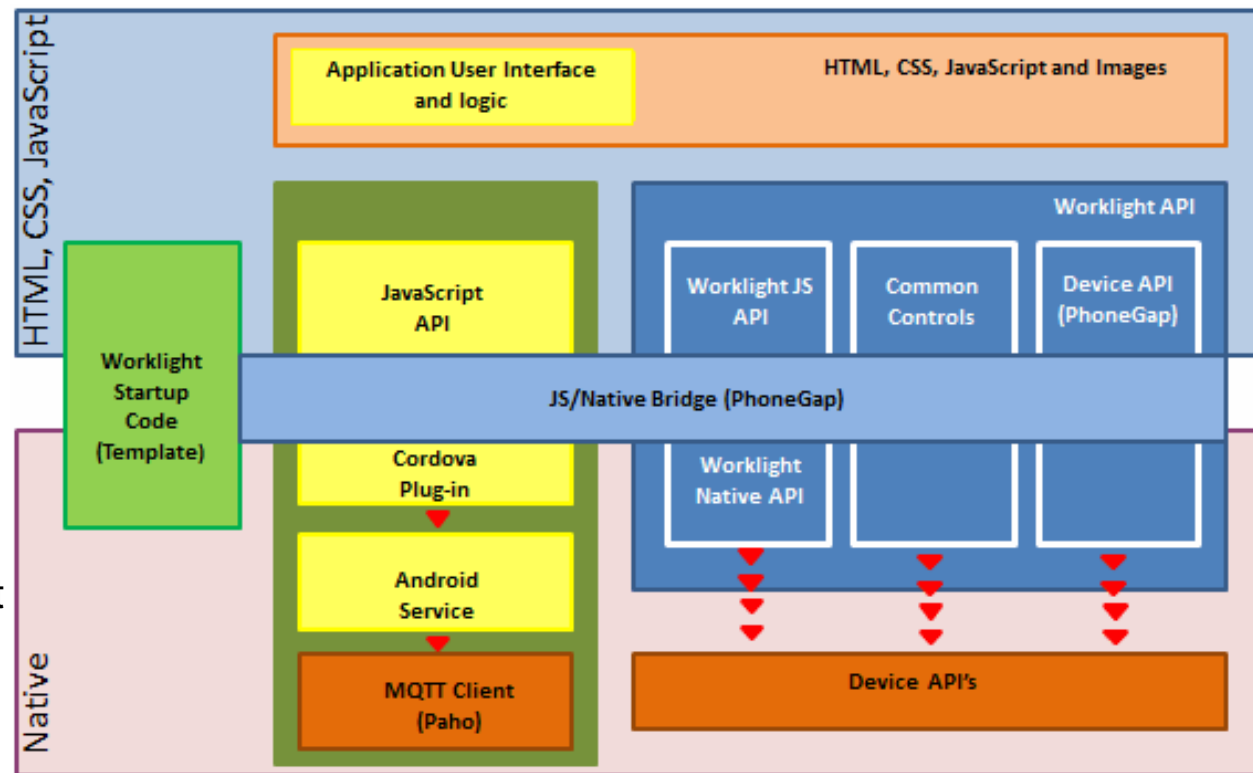
This native code includes an Android Service that manages communications with MQTT when the application itself is not running in the foreground.



# Mobile Hybrid Application

**IBM Worklight Studio can be used to develop a hybrid Android application that sends and receives messages using MQTT.**

The application is developed in HTML and JavaScript and the Apache Cordova (PhoneGap) libraries provided with IBM Worklight are used to interface with native code that implements the MQTT protocol. This native code includes an Android Service that manages communications with MQTT when the application itself is not running in the foreground.



# IBM Worklight

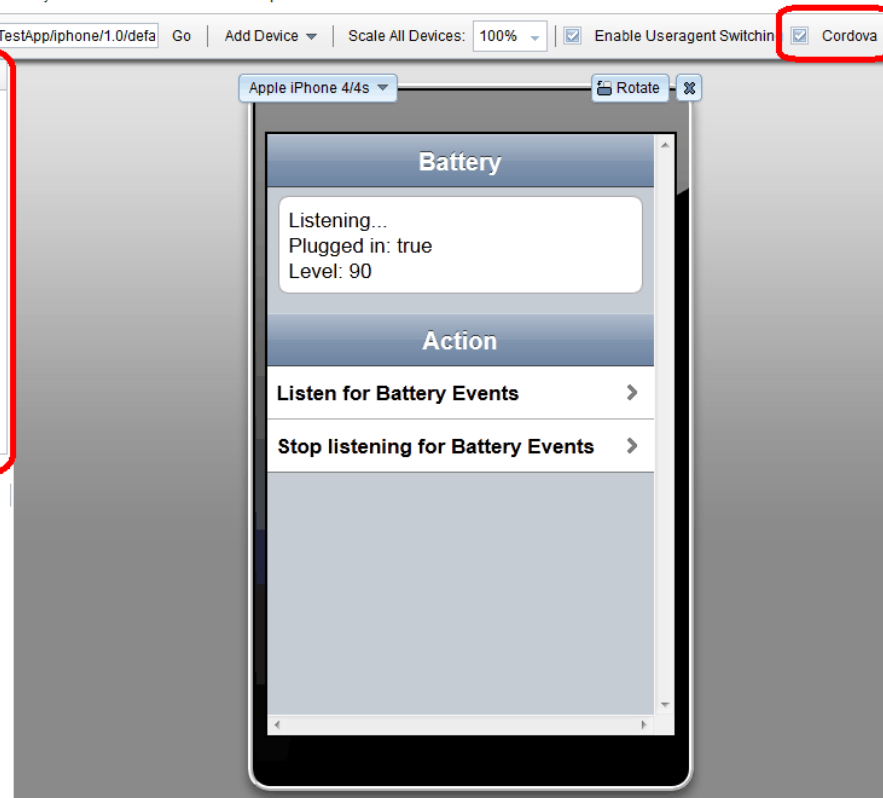
**IBM Worklight Studio offers a mobile browser simulator to test apps using a desktop browser.**

The mobile browser simulator allows users to test hybrid applications leveraging device features without having to run on the actual device. This saves time because the app doesn't have to be repeatedly deployed to devices during testing.

## Mobile Browser Simulator

The Mobile Browser Simulator displays mobile web pages in a variety of mobile browser sizes and shapes.

Webpage: <http://localhost:8080/apps/services/preview/DojoTestApp/iphone/1.0/defa> Go Add Device Scale All Devices: 100%  Enable Useragent Switching  Cordova



# IBM Mobile Messaging and M2M Client Pack MA9B

**MA9B provides an SDK for building MQTT based mobile messaging and machine to machine (m2m) applications.**

You can download the most recent level of the IBM Mobile Messaging and M2M Client Pack MA9B from the IBM developerWorks Messaging website at:

<https://www.ibm.com/developerworks/mydeveloperworks/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/download?lang=en>

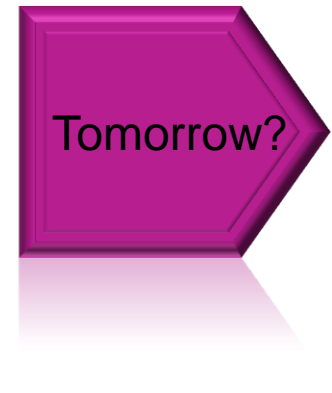
The SDK contains client libraries and samples for use in building mobile and M2M types of applications. This developerworks article walks through using MQTT SDK to create Android app:

[https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/mobile\\_messaging\\_on\\_android?lang=en](https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/mobile_messaging_on_android?lang=en)



# MQTT and MessageSight Now?

## What are the Possibilities for the Future?



- My phone only talked to other phones
- It sent data to others when I asked it to
- phone can connect to almost anything
- It shares and receives info automatically
- What will replace it?
- Who will it talk to?

# Solution Example: Australian Open

Live Grand Slam Tennis Scores

Utilized MessageSight to provide real-time scores with sub-second response time.

30,000 concurrent clients connected

The screenshot shows the 'LIVE SCORES' page for the Australian Open at Rod Laver Arena. The page features a navigation bar with links for 'LIVE SCORES', 'COMPLETED MATCHES', 'IBM SLAMTRACKER', 'SOCIAL LEADERBOARD', and 'RESULTS ARCHIVE'. Below the navigation bar, the page is titled 'LIVE SCORES' and includes a breadcrumb trail: 'HOME > SCORES > MATCHES IN PROGRESS'. The main content area displays a grid of match results for various tennis events, including Men's Singles, Women's Singles, Men's Doubles, Women's Doubles, Mixed Doubles, and Junior Boys' Singles. Each match entry shows the players' names and nationalities, their current score in points and sets, and a 'MATCH STATS' link. For example, in the Men's Singles - Final, R. Nadal (ESP) is leading S. Wawrinka (SUI) with a score of 3-2 in sets and 6-3 in points.

Event	Player 1	Player 2	Pts	1	2	3	4	5
Men's Singles - Final	R. Nadal ESP (1)	S. Wawrinka SUI (6)	3	2	6	3		
			6	6	3	6		
Women's Singles - Final	N. Li CHN (4)	D. Cibulkova SVK (20)	7 <sup>7</sup>	6				
			6 <sup>3</sup>	0				
Men's Doubles - Final	E. Butorac USA R. Klaasen RSA	L. Kubot POL (14) R. Lindstedt SWE (14)	3	3				
			6	6				
Women's Doubles - Final	S. Errani ITA (1) R. Vinci ITA (1)	E. Makarova RUS (3) E. Vesnina RUS (3)	6	3	7			
			4	6	5			
Mixed Doubles - Final	K. Mladenovic FRA D. Nestor CAN	S. Mirza IND (6) H. Tecau ROU (6)	6	6				
			3	2				
Junior Boys' Singles - Final	A. Zverev GER (1)	S. Kozlov USA (2)	6	6				
			3	0				



## Solution Example: Sprint Velocity



Owner pushes start button on app.

Sprint authenticates User

Unlock command sent to connected car, door unlocks

# “Connected City” for better quality of life

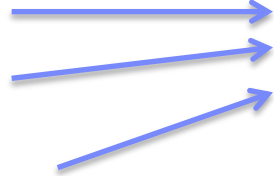


<http://m2m.demos.ibm.com:9080/worklight/apps/services/preview/Traffic/common/0/default/Traffic.html?&ip=75.126.71.163&port=1883&map=DEARBORN>

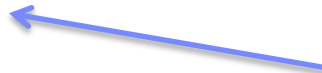




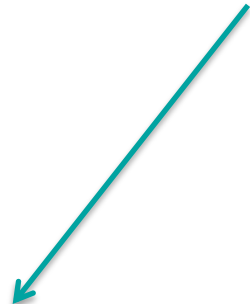
# Connected City Topology/Architecture



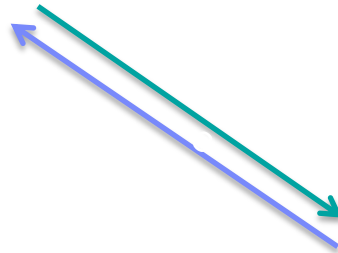
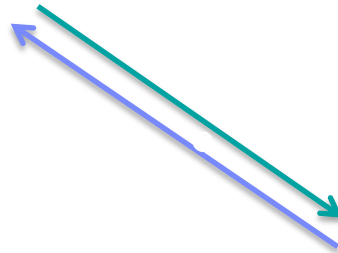
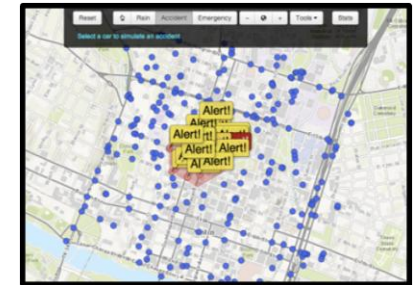
IBM MessageSight



IBM InfoSphere Streams



IBM Worklight





# Retail of the future

- Interact with customers as they go through the store
- Seamlessly integrate online and in-store shopping
- Retail offers at the right time
- Efficient use of sales associates



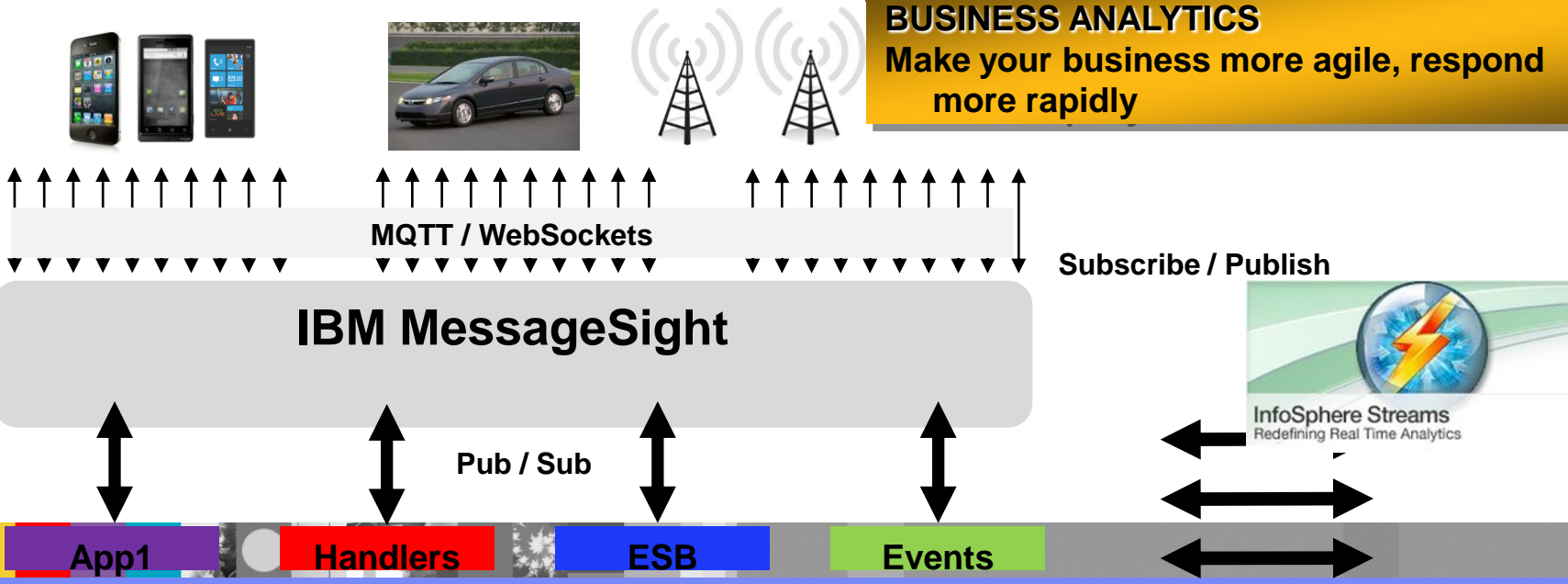
# Messaging Use Cases

**MOBILE**  
 mobile event notification  
 mobile data gathering for business intelligence  
 mobile collaboration  
 mobile payments  
 medical monitoring, automated medical records  
 vehicle telematics, security, routing

**SMARTER CITIES**  
 traffic monitoring and control

**ENERGY/UTILITY**  
 metering (e.g. energy usage, temperature)  
 control (e.g. valves)  
 intelligent/predictive grid control  
 energy broking

**BUSINESS ANALYTICS**  
 Make your business more agile, respond more rapidly



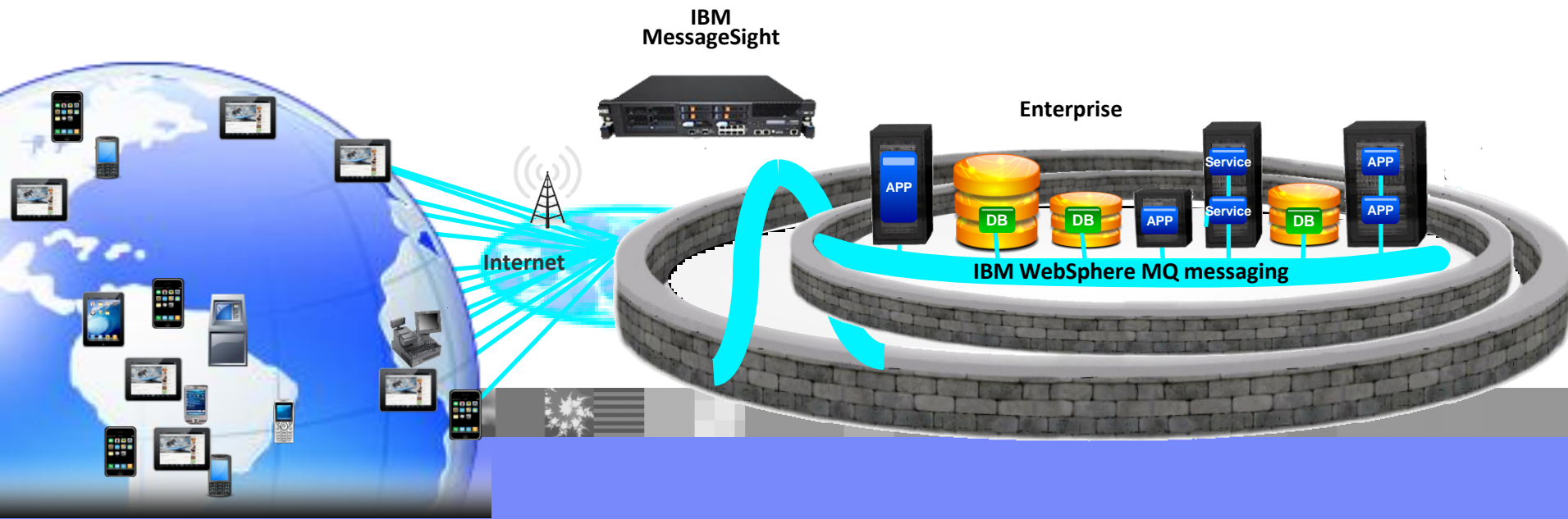
# IBM MessageSight and WebSphere MQ

## IBM MessageSight

- Extends reach beyond enterprise boundary
- Designed for reaching world of mobile devices and smart sensors
- Reliable, low latency delivery
- Scale to handle millions of things, millions of events
- Exploits hardware specific capabilities to optimize throughput

## IBM WebSphere MQ messaging

- Core backbone for enterprise messaging
- Designed for reaching estate of enterprise applications, services, data, processes
- Rich messaging capabilities
- Assured, asynchronous delivery
- Single shared data movement backbone for data, files, events, service interactions
- Reach 80+ platform configurations
- Exploits z/OS platform specific capabilities



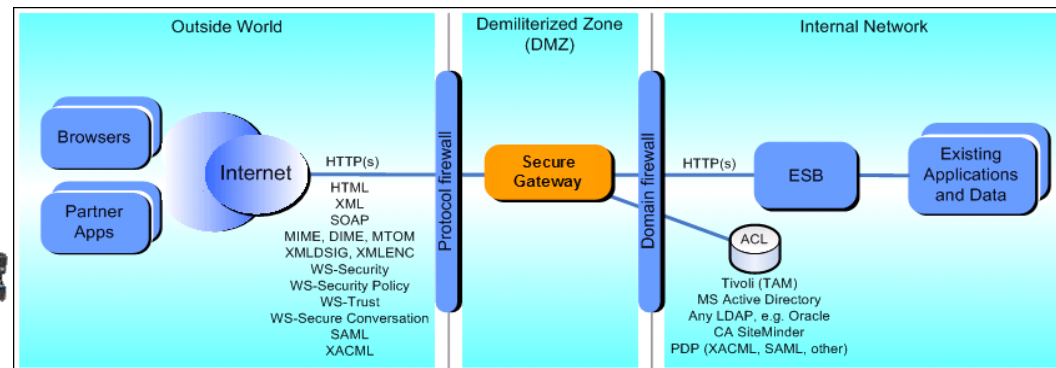
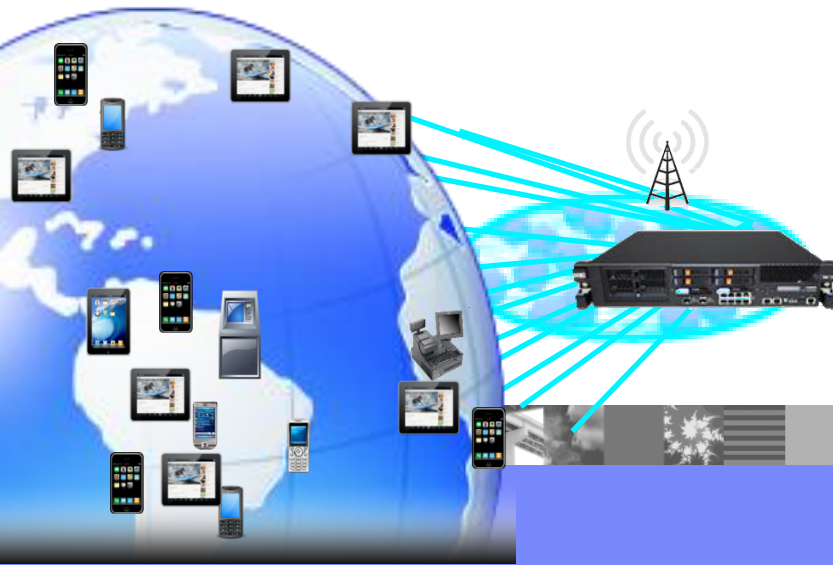
# IBM Messaging Appliance and DataPower

## IBM MessageSight

- Asynchronous messaging provider
- Supports full publish/subscribe messaging with support for MQTT, WebSockets and JMS
- Does not act as a proxy or security gateway
- Scales to millions of things, millions of events
- Replace 1,000 web servers w 1 rack, 273M events / second per rack

## WebSphere DataPower

- Security Gateway for HTTP-based traffic (REST/JSON/XML/SOAP)
- Sits in front of your applications
- Policy Enforcement
- Data Validation
- Message Encryption and Digital Signatures
- Not a messaging provider
- Thousands of synchronous HTTP transactions



## Additional Resource and Useful Links

### MessageSight Infocenter

<http://pic.dhe.ibm.com/infocenter/ism/v1r0m0/index.jsp>

### WebSphere MQ Infocenter

<http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp>

### DeveloperWorks Blog Entry

[https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/ibm\\_messagesight\\_easy\\_to\\_connect\\_mobile](https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/ibm_messagesight_easy_to_connect_mobile)

### MessageSight Development Community

[https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/welcome\\_ibm\\_messaging](https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/welcome_ibm_messaging)

### MQTT.org

[MQTT.org](http://mqtt.org)

### Redbooks

<http://www.redbooks.ibm.com/abstracts/sg248054.html>

<http://www.redbooks.ibm.com/redpieces/abstracts/sg248183.html>

### IBM MobileFirst Connected Car Demo

[https://www.youtube.com/watch?v=v\\_Yf3Qgj5Y0](https://www.youtube.com/watch?v=v_Yf3Qgj5Y0)



# Summary

- **“Internet of Things”**
- **MQTT: low bandwidth protocol utilizing Pub/Sub architecture with QoS for assured delivery**
- ***WMQ Telemetry***
- ***MessageSight connectivity with Mobile Applications and WebSphere MQ***
- ***Examples of MQTT and MessageSight***

# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:  
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>





# Connect with us!

## 1. Get notified on upcoming webcasts

Send an e-mail to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com) with subject line “wste subscribe” to get a list of mailing lists and to subscribe

## 2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)

## 3. Be connected!

Connect with us on [Facebook](#)

Connect with us on [Twitter](#)





# Questions and Answers